AGREEING ON ALLOCATION: SHARED-PREFIX SELECTIVE ADAPTATION FOR FROZEN SPEECH CODECS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural speech codecs typically allocate a uniform code budget across all frames. Adaptive frame-wise allocation can improve coding efficiency but often requires retraining or extra signaling. We introduce Agreeing on Allocation, a post-hoc method that distributes a fixed code budget in frozen codecs. Using the mandatory first-stage residual vector quantization (RVQ) codes as a shared prefix, the encoder and decoder independently recover the same budget-constrained stage assignment without frame-wise signaling. However, agreement alone does not ensure better reconstruction. We therefore add selective deployment, in which the encoder compares proposed and reference reconstructions at the same budget using fidelity and perceptual criteria and signals only an utterance-level choice. For the fixed selection rule, held-out evaluation provides a finite-sample upper confidence bound on the probability of a quality drop beyond a predefined margin relative to the reference. Objective evaluations show codec-dependent quality gains at matched budgets, supported by descriptive listening results. Selective deployment reduces observed PESQ-drop events, with pointwise risk bounds under the stated cluster-sampling assumptions.

1 INTRODUCTION

Efficient speech coding depends on how the available bit budget is allocated across the signal (Shoham & Gersho, 1988). Adaptive allocation can improve reconstruction, but communicating the allocation consumes part of that budget (Brandenburg & Stoll, 1994). This raises the question of whether useful allocations can be derived from information already shared by the encoder and decoder, without explicit allocation signaling.

Neural speech codecs such as SoundStream and DAC (Zeghidour et al., 2022; Kumar et al., 2023) use residual vector quantization (RVQ) to represent each frame with discrete codes from successive stages. They support different rates by varying the number of active RVQ stages, typically using the same number at every frame within an operating point. This uniform allocation can be inefficient because the benefit of additional stages varies across frames (Chae et al., 2025; Zhao et al., 2026). VRVQ (Chae et al., 2025) learns frame-wise allocation jointly with the codec. We instead study offline, utterance-level reallocation of a fixed total code budget, training only post-hoc allocation modules while keeping the pretrained encoder, decoder, and RVQ codebooks frozen.

When the number of active RVQ stages varies across frames, the decoder needs the allocation to parse later-stage codes. Post-hoc methods such as BAMU (Zhao et al., 2026) use encoder-side representations to estimate allocation utilities and transmit the resulting allocation through a compressed map. Avoiding this signaling requires both sides to derive the same allocation from shared information. Yet later RVQ stages encode information left unresolved by earlier stages (Zeghidour et al., 2022; Kumar et al., 2023), so their benefits may be harder to predict from a shared coarse representation than from richer encoder-side features. The question is how much allocation benefit can be retained under this information constraint.

Encoder–decoder agreement does not ensure that a learned allocation improves reconstruction over the same-budget reference. Inaccurate utility estimates can cause quality regressions on individual inputs. Selective prediction (Chow, 1970; Geifman & El-Yaniv, 2019) and learning to defer (Mozannar & Sontag, 2020) motivate choosing between a learned proposal and a reference policy rather

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

than deploying the proposal unconditionally. However, the quality criterion used to assess regressions may not be directly observable at deployment. Available fidelity and perceptual indicators can mis-rank the reconstructions, leaving residual risk that requires separate evaluation.

We introduce Agreeing on Allocation for frozen RVQ codecs. The mandatory first-stage RVQ codes form a shared prefix from which both sides estimate additional-stage utilities and recover the same allocation through deterministic budget-constrained optimization, without a frame-wise map. The encoder compares the proposed and same-budget reference reconstructions using fidelity and perceptual criteria, deploying the proposal only when both criteria pass and otherwise retaining the reference. Only an utterance-level choice is signaled.

Our contributions are twofold:

- **Shared-prefix allocation.** We enable fixed-budget frame-wise reallocation in frozen RVQ codecs using mandatory first-stage codes, without transmitting an allocation map.
- **Selective deployment and risk evaluation.** We combine proposal selection with same-budget reference fallback and report a finite-sample upper confidence bound on the fixed policy's quality-drop risk using reserved confirmation cohorts whose outcomes are not used to retune the evaluated policy (Clopper & Pearson, 1934).

Objective evaluations and subjective listening tests show codec-dependent reconstruction gains at fixed code budgets. Selective deployment reduces observed PESQ regressions while retaining part of these gains. Coordination experiments quantify the trade-off among reconstruction quality, signaling overhead, and decoder-side computation.

2 Related Work

Neural audio codecs. Classical multistage vector quantization and embedded speech coding established progressive refinement across coding layers (Juang & Gray, 1982; Ramprashad, 2000). Neural codecs such as SoundStream and DAC (Zeghidour et al., 2022; Kumar et al., 2023) use residual vector quantization (RVQ) for coarse-to-fine reconstruction, with quantizer dropout supporting multiple rates through different numbers of active RVQ stages. At a given operating point, these codecs typically use a uniform number of RVQ stages across frames. We instead learn an allocator that redistributes a fixed code budget while keeping the encoder, decoder, and RVQ codebooks frozen.

Adaptive allocation and coordination. Content-dependent rate allocation is well established in perceptual audio coding (Johnston, 1988) and learned image compression (Ballé et al., 2018). VRVQ jointly learns frame-wise RVQ stage counts with the codec (Chae et al., 2025), with a noise-robust extension targeting allocation under noisy inputs (Chae & Lee, 2025). BAMU performs post-hoc utility-based allocation for frozen codecs but transmits a compressed depth map (Zhao et al., 2026), whereas Gogo allocates token budgets to coarse-to-fine groups (Chen et al., 2026). Other approaches vary temporal resolution through frame selection, merging, or clustering (Zhang et al., 2025; Wang et al., 2026; Li et al., 2026; Zheng et al., 2026).

Coordination can rely on explicit side information, including keep masks (Sohn & Nam, 2026), segment lengths (Bralios et al., 2026), and durations (Zheng et al., 2026). UniStream (Zhao & Wu, 2026) instead recovers expert-codebook routing from previously decoded quantized states without transmitting expert identifiers, but jointly trains a streaming codec. Our method performs post-hoc frame-wise RVQ stage allocation in frozen codecs, recovering both proposal and reference allocations from mandatory first-stage codes, the budget, and identical fixed parameters without an allocation map.

Reliability and selective deployment. Selective prediction allows abstention on selected inputs (Chow, 1970; Geifman & El-Yaniv, 2019), while learning to defer routes inputs to another predictor, including through post-hoc deferral for a fixed model (Mozannar & Sontag, 2020; Narasimhan et al., 2022). Risk-controlling prediction sets, Learn then Test, and Conformal Risk Control use calibration data to obtain risk guarantees under their respective assumptions (Bates et al., 2021; Angelopoulos et al., 2025; 2024). Selective Conformal Risk Control addresses risk over accepted predictions (Xu et al., 2025).

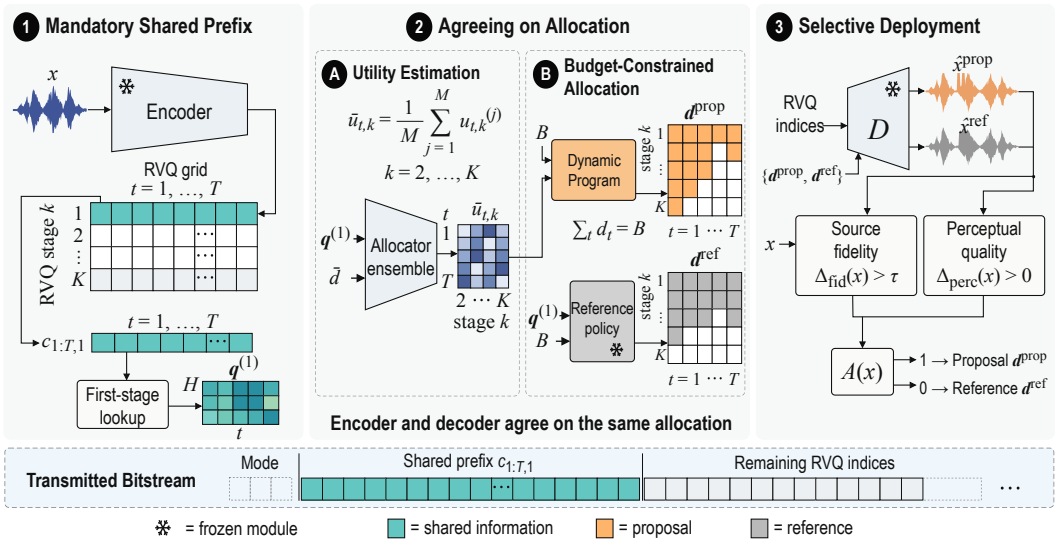

Figure 1: Overview of Agreeing on Allocation. Mandatory first-stage RVQ codes provide a shared prefix for utility estimation and deterministic allocation under a fixed code budget. The encoder and decoder recover identical allocations without a frame-wise map. The encoder compares proposal and reference reconstructions using fidelity and perceptual criteria, conveying its choice through an utterance-level mode.

Our setting uses reference fallback rather than abstention, retaining a reconstruction for every input at the same code budget. Instead of calibrating a policy to meet a target risk, we evaluate a selection rule fixed independently of the risk-evaluation data. Held-out utterance clusters provide a finite-sample upper confidence bound on the probability of a predefined quality drop, measured over all clusters rather than conditional on acceptance (Section 3.5).

3 Method

Figure 1 summarizes the proposed method. We formulate the offline utterance-level allocation problem, describe shared-prefix utility estimation, budget-constrained optimization, and encoder–decoder coordination, then introduce selective deployment and evaluate the fixed policy's quality-drop risk.

3.1 Problem Formulation

We consider a frozen neural codec with encoder E, decoder D, and K RVQ stages. Given a mono waveform $x \in \mathbb{R}^L$ of L samples, the encoder produces a latent representation $E(x) \in \mathbb{R}^{H\times T}$, where H is the number of channels and T is the number of frames, with the batch dimension omitted for simplicity. For frame $t \in \{1,\dots,T\}$ and stage $k \in \{1,\dots,K\}$, let $c_{t,k}$ denote the integer code index. We denote the corresponding RVQ embedding after any output projection by $e_k(c_{t,k}) \in \mathbb{R}^H$. An allocation $\mathbf{d} = (d_1,\dots,d_T)$ retains the first $d_t \in \{1,\dots,K\}$ RVQ stages at frame t, yielding

$$\widetilde{z}_t(\mathbf{d}) = \sum_{k=1}^{d_t} e_k(c_{t,k}), \qquad \widehat{x}_{\mathbf{d}} = D(\widetilde{z}_{1:T}(\mathbf{d}))\,. \tag{1}$$

Here, $\widetilde{z}_{1:T}(\mathbf{d}) \in \mathbb{R}^{H\times T}$ stacks the frame vectors as columns. For an integer budget B counting transmitted RVQ indices, with $T \le B \le TK$, the feasible allocations are

$$\mathcal{D}(B) = \left\{ \mathbf{d} \in \{1,\dots,K\}^T : \sum_{t=1}^{T} d_t = B \right\}, \qquad \bar{d} = \frac{B}{T}, \tag{2}$$

where $\bar{d}$ is the mean number of active stages per frame. Since $d_t \geq 1$ for every frame, the first-stage indices $c_{1:T,1}$ are always transmitted as part of the reconstruction payload and are therefore available to both the encoder and decoder. These indices form the shared prefix used for allocation.

At each budget B, let $d^{\mathrm{ref}} \in \mathcal{D}(B)$ denote the allocation produced by a fixed codec-specific reference policy, denoted by Ref. Unlike the uniform-depth CBR baseline, Ref uses a separately fitted codec-specific reference cost predictor that is frozen during training and inference of the proposed allocator. We construct a proposal $d^{\mathrm{prop}} \in \mathcal{D}(B)$ by learning corrections to these reference utility scores, aiming to improve reconstruction quality at the same budget. To avoid frame-wise signaling, both allocations must be independently recoverable at the encoder and decoder from the shared prefix, the budget B, and identical fixed parameters. The codec-specific reference policies are detailed in Appendix A.1.

3.2 Shared-Prefix Allocation

From the shared first-stage indices $c_{1:T,1}$, the encoder and decoder reconstruct the first-stage quantized latent

$$q_t^{(1)} = e_1(c_{t,1}), \qquad \mathbf{q}^{(1)} = [q_1^{(1)}, \ldots, q_T^{(1)}] \in \mathbb{R}^{H \times T}. \tag{3}$$

Given $\mathbf{q}^{(1)}$ and the operating point $\bar{d}$, we compute fixed marginal reference scores $u_{t,k}^{\mathrm{ref}}$ for $k = 2, \ldots, K$. Each of M independently trained budget-conditioned temporal convolutional allocators a_{θ_j} takes $(\mathbf{q}^{(1)}, \bar{d})$ as input and predicts a bounded correction $\Delta u_{t,k}^{(j)}$:

$$u_{t,k}^{(j)} = u_{t,k}^{\mathrm{ref}} + \Delta u_{t,k}^{(j)}, \qquad j = 1, \ldots, M, \quad k = 2, \ldots, K. \tag{4}$$

We average the corrected utilities across models:

$$\bar{u}_{t,k} = \frac{1}{M} \sum_{j=1}^{M} u_{t,k}^{(j)}. \tag{5}$$

The proposed allocation is

$$\mathbf{d}^{\mathrm{prop}} = \arg \max_{\mathbf{d} \in \mathcal{D}(B)} \left[\sum_{t=1}^{T} \sum_{k=2}^{d_t} \bar{u}_{t,k} - \lambda \sum_{t=2}^{T} |d_t - d_{t-1}| \right], \tag{6}$$

where $\lambda \geq 0$ penalizes changes in RVQ stage count between adjacent frames. We solve Eq. (6) exactly by dynamic programming under budget B.

Each allocator is trained using quality scores of complete candidate allocations rather than directly observed stage-level gains. Omitting the model index j, the predicted score of a candidate $\mathbf{d} \in \mathcal{D}(B)$ is

$$s_\theta(\mathbf{d}) = \sum_{t=1}^{T} \sum_{k=2}^{d_t} u_{t,k}. \tag{7}$$

The first-stage term is omitted because it is constant across candidates for a given input. With the codec frozen, candidate reconstructions are scored against clean training targets using a fixed quality metric. The allocator is trained with listwise ranking and regression losses, with predicted and target scores normalized across candidates for each input. The additive score is a surrogate for comparing complete allocations rather than an exact decomposition of reconstruction quality, and the smoothness term in Eq. (6) is applied only during allocation. Appendices A.1 and A.2 provide details of the reference policy, allocator training, and dynamic-programming optimization.

3.3 Encoder–Decoder Coordination

The decoder must recover the frame-wise allocation before parsing later-stage RVQ indices. We therefore transmit all first-stage indices $c_{1:T,1}$ before the later-stage payload. With the frame count T and operating point $\bar{d}$ in the header, the decoder reconstructs $\mathbf{q}^{(1)}$ and obtains $B = T\bar{d}$ without knowing the allocation.

An utterance-level mode in the header identifies the selected allocation. In proposal mode, the decoder evaluates the fixed allocator ensemble on $(\mathbf{q}^{(1)}, \bar{d})$ and solves Eq. (6). In reference mode, it

applies the fixed codec-specific policy to the same shared inputs. The recovered depth d_t specifies how many later-stage indices $(d_t - 1)$ to read for frame t, removing the need for a frame-wise allocation map.

Both sides use identical fixed parameters and deterministic procedures for shared-prefix reconstruction, utility evaluation, and allocation, including tie-breaking. Under these conditions, the shared prefix, budget, and mode uniquely determine the allocation at both ends (Appendix A.3). Agreement ensures consistent parsing, but not better reconstruction, motivating selective deployment.

3.4 Selective Deployment

At the encoder, the frozen decoder reconstructs $\widehat{x}_{\mathbf{d}^{\mathrm{prop}}}$ and $\widehat{x}_{\mathbf{d}^{\mathrm{ref}}}$ at the same code budget. We compare their fidelity to the codec input x and their predicted perceptual quality. Let $\Delta_{\mathrm{fid}}(x)$ denote the reference-minus-proposal difference in multi-resolution Mel distortion relative to x, normalized by a fixed positive scale for the operating point. Let $\Delta_{\mathrm{perc}}(x)$ denote the proposal-minus-reference difference in UTMOS (Saeki et al., 2022), a no-reference perceptual quality score. Both are positive when they favor the proposal. For each operating point, we fix a threshold $\tau \geq 0$ and accept the proposal only when both criteria pass:

$$A(x) = \mathbf{1}\left[\Delta_{\mathrm{fid}}(x) > \tau \ \wedge \ \Delta_{\mathrm{perc}}(x) > 0\right]. \tag{8}$$

Setting $\tau = +\infty$ always selects the reference. The deployed allocation is

$$\mathbf{d}^{\mathrm{out}}(x) = \begin{cases} \mathbf{d}^{\mathrm{prop}}(x), & A(x) = 1, \\ \mathbf{d}^{\mathrm{ref}}(x), & A(x) = 0. \end{cases} \tag{9}$$

The encoder signals $A(x)$ through the utterance-level mode, and the decoder recovers the selected allocation as described in Section 3.3. Score definitions, normalization, and threshold selection are detailed in Appendix B.1.

3.5 Quality-Drop Risk Evaluation

Selection does not require clean targets; risk evaluation does. For a fixed codec and operating point, let $y \in \mathbb{R}^L$ be the clean target associated with input x, and let $Q(y, \widehat{x})$ be a prespecified reference-based quality metric for which larger values indicate better quality. For a prespecified margin $m > 0$, define

$$\mathcal{E}(x, y) = \mathbf{1}\left[A(x) = 1 \ \wedge \ Q(y, \widehat{x}_{\mathbf{d}^{\mathrm{prop}}}) - Q(y, \widehat{x}_{\mathbf{d}^{\mathrm{ref}}}) < -m\right]. \tag{10}$$

This event occurs when an accepted proposal loses more than m in quality relative to the reference.

Versions of the same utterance under predefined acoustic conditions $\mathcal{C}$ form a speech cluster $Z = \{(x_\ell, y_\ell)\}_{\ell \in \mathcal{C}}$. Selection is applied separately to each input, and a cluster incurs an event if any condition does:

$$\mathcal{E}_{\mathrm{cl}}(Z) = \max_{\ell \in \mathcal{C}} \mathcal{E}(x_\ell, y_\ell). \tag{11}$$

For the target cluster distribution $\mathcal{P}_{\mathrm{cl}}$, the population risk is

$$R = \Pr_{Z \sim \mathcal{P}_{\mathrm{cl}}}\left[\mathcal{E}_{\mathrm{cl}}(Z) = 1\right]. \tag{12}$$

This risk is measured over all clusters, not conditional on proposal acceptance.

The deployment policy and event definition are fixed independently of the evaluation data. Let $Z_1, \ldots, Z_n$ be i.i.d. draws from $\mathcal{P}_{\mathrm{cl}}$, and let $r = \sum_{i=1}^{n} \mathcal{E}_{\mathrm{cl}}(Z_i)$ count clusters with an event. For $\delta \in (0, 1)$, the one-sided Clopper–Pearson upper confidence bound (Clopper & Pearson, 1934) is

$$U_{\mathrm{CP}}(r, n, \delta) = \begin{cases} \mathrm{Beta}^{-1}(1 - \delta; r + 1, n - r), & r < n, \\ 1, & r = n, \end{cases} \tag{13}$$

where $\mathrm{Beta}^{-1}(p; \alpha, \beta)$ is the p-quantile of the beta distribution with shape parameters α and β. Under these assumptions,

$$\Pr\left[R \leq U_{\mathrm{CP}}(r, n, \delta)\right] \geq 1 - \delta. \tag{14}$$

with probability over draws of the evaluation clusters. Cluster construction, evaluation splits, and bound computation are detailed in Appendix B.2.

Table 1: Proposal gains on 1,000 LibriSpeech test-clean utterances at matched RVQ-index budgets, without selective deployment. Positive values favor Proposal.

Codec	$\bar{d}$	Proposal − CBR		Proposal − Ref	
		PESQ	ViSQOL	PESQ	ViSQOL
DAC	2	0.166	0.117	0.072	0.051
	4	0.130	0.070	0.034	0.022
	6	0.063	0.008	0.053	0.008
SpeechTokenizer	2	0.129	0.176	0.028	0.035
	4	0.086	0.159	0.023	0.045
	6	0.015	0.033	0.006	0.004
PhoenixCodec	2	0.181	0.379	0.043	0.018
	3	0.202	0.369	0.039	−0.009
	4	0.122	0.142	0.021	−0.007
	5	0.045	0.061	0.007	−0.010

4 Experiments

4.1 Experimental Setup

Data and codecs. We use LRAC challenge data (Cisco Open Source, 2026) for allocator training, checkpoint selection, and held-out risk evaluation, and the official LRAC 2025 Track 1 public test set for coordination and external-baseline evaluation. We additionally evaluate allocation and selective deployment on 1,000 LibriSpeech test-clean utterances (Panayotov et al., 2015). Because source metadata identifies shared chapters with the LRAC source pool, this benchmark is descriptive and separate from held-out risk evaluation. We evaluate frozen DAC (Kumar et al., 2023), SpeechTokenizer (Zhang et al., 2024), and our reproduced PhoenixCodec (Wan et al., 2026). Data splits, codec configurations, and resampling procedures are provided in Appendices C.1 and C.2.

Operating points and budgets. We use $K = 8$ RVQ stages with $\bar{d} \in \{2, 4, 6\}$ for DAC and SpeechTokenizer, and $K = 6$ with $\bar{d} \in \{2, 3, 4, 5\}$ for PhoenixCodec. Proposal averages utilities from three independently seeded allocators ($M = 3$), with $\lambda = 0.1$. Table 1 matches exactly $B = T\bar{d}$ RVQ indices per utterance and codec. Coordination comparisons instead match total transmitted bits, including RVQ payload, allocation signaling, and serialization overhead. Training settings, including optimizer and checkpoint selection, and bit accounting are detailed in Appendix C.2.

Baselines and controls. CBR uses a uniform RVQ depth at the requested operating point. Ref uses the frozen codec-specific reference policy described in Section 3.1. Proposal uses the learned shared-prefix allocation without selective deployment, while Selective chooses between Proposal and Ref using the selection rule in Section 3.4. Our BAMU (Zhao et al., 2026) implementation and Gogo-based SP-Count (Chen et al., 2026) test alternative allocation strategies. Encoder-LoRA (Hu et al., 2022) tests encoder adaptation with RVQ and decoder frozen; VRVQ (Chae et al., 2025) tests jointly trained variable-rate coding on a matched backbone. FlexiCodec provides supplementary system-level context. Protocols and implementation differences appear in Section 4.3 and Appendices D.6 and D.7. Random@Acceptance uses 1,000 matched-count draws; single-criterion selectors isolate the two gates. Compressed-Map tests explicit signaling with the same utility predictor but a different solver (Appendix D.3).

Metrics and statistics. Main results report PESQ (Rix et al., 2001), ViSQOL (Chinen et al., 2020), and listening scores. PESQ is training-aligned and UTMOS is selection-aligned, so neither alone establishes perceptual improvement. Test-clean contrasts use pointwise 95% paired speaker-bootstrap intervals; external protocols and their resampling units are specified in the appendix. Held-out evaluation follows Section 3.5, with PESQ margin $m = 0.05$ and policies fixed without using the confirmation outcomes. We report cluster event counts, pointwise one-sided 95% Clopper–Pearson upper bounds, and input-level acceptance. All six objective metrics, preprocessing, and statistical procedures are detailed in Appendix C.4.

Table 2: Alternative adaptation comparisons. (a) Matched bytes at $\bar{d} = 4$: LoRA PESQ differences on test-clean and DAC scores on LRAC public test; LoRA and BAMU entries average three runs. (b) LRAC matched-backbone PESQ differences using a separate spectral-supervised controller, with signaling-inclusive rates and interpolated VRVQ scores. Differences are first minus second.

(a) Encoder adaptation and allocation

Codec	LoRA − Prop.	LoRA − Sel.
DAC	-0.1287	-0.1043
SpeechTokenizer	+0.0891	+0.1031
PhoenixCodec	-0.1869	-0.1682
Method	**PESQ**	**ViSQOL**
BAMU (reproduced)	2.0531	2.6431
Shared-Prefix	2.1379	2.6690

(b) Variable-rate coding

	Post-hoc minus	
Rate (kbps)	CBR	VRVQ
1.5405	+0.0558	+0.2429
3.0417	+0.0015	+0.0748
4.5437	-0.0057	-0.0016

4.2 Fixed-Budget Reallocation

We compare Proposal with CBR and the codec-specific Ref at exactly $B = T\bar{d}$ RVQ indices per utterance. This test-clean benchmark is descriptive and separate from held-out risk evaluation (Section 4.1). Across the ten settings in Table 1, Proposal improves mean PESQ and ViSQOL over CBR and mean PESQ over Ref. For DAC and SpeechTokenizer at $\bar{d} \in \{2, 4\}$, gains over Ref hold for both metrics, with pointwise 95% paired speaker-bootstrap intervals excluding zero. These results show that the complete Proposal design can improve over both uniform allocation and the codec-specific reference at the same RVQ-index budget. Relative to Ref, the observed gains reflect the learned utility correction together with the Proposal allocation procedure, and should not be attributed to the correction alone.

The gains remain codec- and metric-dependent. PhoenixCodec's PESQ gain over Ref decreases with budget, while its ViSQOL changes are negative at $\bar{d} \in \{3, 4, 5\}$. At $\bar{d} = 6$, the ViSQOL interval includes zero for DAC, as do both metric intervals for SpeechTokenizer. Thus, the Proposal provides additional gains over Ref in several settings, but the improvements are not uniform across codecs or quality metrics. Six-metric Proposal–Ref contrasts and pointwise intervals appear in Appendix D.1.

4.3 Comparison with Alternative Adaptation Methods

Table 2 compares encoder adaptation, utility-based allocation, and jointly trained variable-rate coding under their respective budget-matching protocols.

Allocation versus encoder adaptation. Shared-Prefix exceeds all three BAMU runs under our matched-data protocol, with PESQ/ViSQOL of 2.1379/2.6690 versus BAMU averages of 2.0531/2.6431. It also avoids a transmitted allocation map, although differences in supervision and optimization prevent isolating the effect of signaling removal.

Encoder-LoRA adapts the sender while retaining the original RVQ codebooks and decoder. It trails Selective on DAC and PhoenixCodec but leads on SpeechTokenizer at every tested rate. At $\bar{d} = 4$, its PESQ differences from Selective are -0.1043, -0.1682, and $+0.1031$, respectively. The same ordering holds against ungated Proposal, so it is not solely an effect of reference fallback. These results support allocation as an alternative when codec weights must remain fixed, not universal superiority over encoder adaptation.

The Gogo-based SP-Count control also trails Proposal in raw PESQ gain at all three DAC rates, although it is slightly better after gating at $\bar{d} = 2$. This motivates evaluating allocation and selection separately: a pre-gating advantage need not persist afterward. Implementation differences, all LoRA operating points, and seed variability appear in Appendices D.6 and D.7.

Jointly trained variable-rate coding. The paired 24-kHz CBR/VRVQ backbones test shared-prefix adaptation beyond the main pretrained codecs. A separate spectral-supervised post-hoc controller improves PESQ over rate-interpolated VRVQ by 0.2429 and 0.0748 at the two lower bitrates,

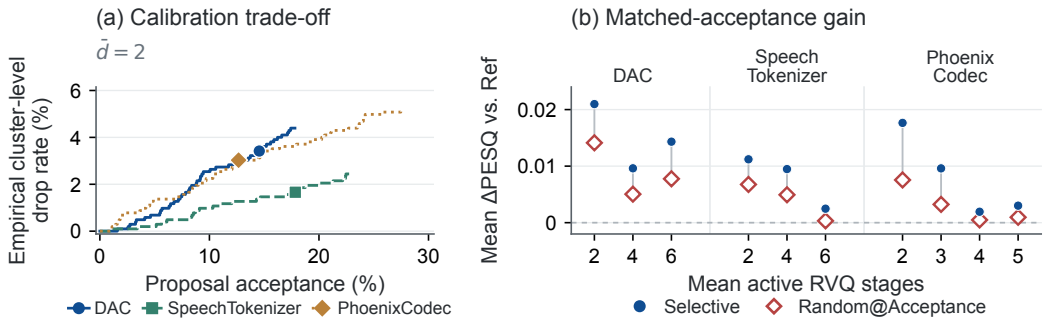

Figure 2: Selective deployment. (a) Calibration sweeps at $\bar{d} = 2$ with the UTMOS gate fixed; markers denote frozen thresholds. Rates are empirical, not risk bounds. (b) Mean PESQ gains over Ref on 1,000 test-clean inputs. Random@Acceptance averages 1,000 matched-count draws; segments connect means, not uncertainty limits.

Table 3: Selector ablations at $\bar{d} = 4$ on 1,000 descriptive test-clean inputs. A: acceptance (%); G: mean PESQ gain over Ref across all inputs; D: input-level quality-drop rate (%, margin 0.05), not cluster-level risk.

Selector	DAC			SpeechTokenizer			PhoenixCodec		
	A	G	D	A	G	D	A	G	D
Fidelity	17.4	0.0114	0.2	29.4	0.0122	0.2	3.6	0.0026	0.3
Perceptual	81.0	0.0289	1.6	65.1	0.0204	1.7	52.7	0.0146	3.4
Selective	14.9	0.0096	0.2	21.1	0.0095	0.2	2.3	0.0019	0.0

including map overhead. Its corresponding differences from CBR are 0.0558 and 0.0015; at the highest rate, they become -0.0057 against CBR and -0.0016 against VRVQ. The highest-rate VRVQ comparison has a paired interval containing zero, indicating that the observed advantage is concentrated at lower rates.

VRVQ also trails its paired CBR at the lower rates, so the larger gains over VRVQ partly reflect this adaptation's performance. Rate interpolation compares quality at matched measured bitrates rather than newly decoded operating points, and the spectral-supervised controller differs from the main ensemble. This provides complementary evidence for shared-prefix allocation on a matched backbone, not a general ranking of published systems. Protocols, intervals, and FlexiCodec results appear in Appendix D.6; coordination quality and cost appear in Appendix D.3.

4.4 Selective Deployment and Residual Risk

Selection Beyond Reduced Acceptance. Figure 2(a) shows calibration acceptance–drop trade-offs for DAC, SpeechTokenizer, and PhoenixCodec. Stricter thresholds reduce unconditional event rates by construction, so this trend alone does not establish informative selection. On the descriptive test-clean benchmark, Selective achieves higher mean PESQ gain than the average random control at all ten displayed settings with matched acceptance (Figure 2(b)). This supports informed selection beyond reduced proposal usage alone. Section 4.6 separately examines the contribution of each selection criterion.

Residual Risk Under Frozen Policies. On the separate confirmation splits, the ten settings in Table 4 accept 9.18%–21.84% of inputs and retain positive mean PESQ gains over Ref. Cluster event counts fall from 248–392 under Proposal-Only to 21–41 under Selective, out of 1,024 clusters per setting. This reduction relies on substantial reference fallback; pointwise 95% upper bounds of 2.94%–5.16% quantify the remaining quality-drop risk. Under the assumptions in Section 3.5, these bounds concern unconditional cluster-level risk, not harm conditional on acceptance or subjective preference, and do not establish satisfaction of a prespecified risk target. Additional selector comparisons and statistical details appear in Appendix D.4.

Table 4: Frozen-policy confirmation results: 1,024 clusters and 3,072 inputs per setting. Events follow Eq. (11) with $m = 0.05$. Gain over Ref and acceptance are averaged over all inputs. $U_{0.95}$ is the pointwise one-sided 95% Clopper–Pearson upper bound under i.i.d. clusters, without simultaneous coverage.

Codec	$\bar{d}$	Proposal-Only	Selective			
		Events	ΔPESQ	Accept (%)	Events	$U_{0.95}$ (%)
DAC	2	363	0.008	14.58	38	4.84
	4	272	0.004	12.76	25	3.39
	6	280	0.004	15.23	41	5.16
SpeechTokenizer	2	248	0.010	17.64	24	3.28
	4	300	0.005	14.23	30	3.95
	6	256	0.003	9.18	21	2.94
PhoenixCodec	2	328	0.016	13.22	31	4.06
	3	392	0.020	13.41	32	4.17
	4	385	0.014	9.80	33	4.29
	5	269	0.014	21.84	39	4.95

4.5 Subjective Evaluation

We conduct a blinded, randomized MUSHRA-style test at $\bar{d} \in \{2, 4\}$ with hidden references and low-pass anchors. The 35 completed non-QA sessions provide 140 ratings per method per codec, with reconstructions matched in total bits. DAC uses LRAC public clean speech; SpeechTokenizer and PhoenixCodec use LibriSpeech test-clean. Comparisons are within codec.

Proposal gains 2.18–2.51 mean points over Ref (Figure 3). All completed non-QA sessions are retained; frequent low hidden-reference ratings limit this descriptive evidence. Full results and diagnostics appear in Appendix F.

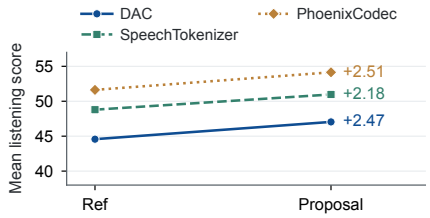

Figure 3: Listening scores and mean gains over Ref.

4.6 Ablations and Analysis

Selection criteria. Compared with perceptual-only selection, the conjunction reduces input-level PESQ drops from 1.6–3.4% to 0–0.2%, at lower acceptance and mean gain (Table 3). Relative to fidelity-only selection, it adds positive UTMOS gain but lowers PESQ drops only for PhoenixCodec; matched-acceptance results appear in Figure 2(b).

Allocator design. On 1,021 reused DAC calibration clusters at $\bar{d} = 4$, removing smoothing increases the cluster-level PESQ-drop rate from 25.27% to 35.95%, while mean gain changes from 0.0105 to 0.0115 and its paired difference interval includes zero. The ensemble also does not outperform its strongest observed member (gain 0.0119; drop rate 23.31%). Full results appear in Appendix D.5.

5 Conclusion

Agreeing on Allocation uses shared first-stage codes to reallocate fixed RVQ budgets in frozen codecs without frame-wise signaling. Experiments show codec-dependent quality gains, while selective deployment improves PESQ over matched random acceptance. Held-out evaluation characterizes residual PESQ-drop risk under stated sampling assumptions. Future work will reduce receiver computation and strengthen independent perceptual validation.

AI USE STATEMENT

Generative AI tools, including OpenAI ChatGPT and Codex, were used for language polishing, literature discovery and organization, and limited assistance with code editing and debugging. All AI-assisted text and code were reviewed by the authors, and all reported experiments, analyses, and claims were independently verified. Generative AI was not relied upon to establish the statistical guarantees reported in the paper; their formulation, assumptions, derivations, and scope were independently verified by the authors. The authors take full responsibility for the content and results presented in this work.

ETHICS STATEMENT

Our subjective listening study collected 35 completed non-QA sessions through an anonymous public link. All participants received a study description and explicitly provided informed consent before taking part. Participation was voluntary. The listening materials were used in accordance with the applicable licenses and permitted uses.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of our results. The anonymous supplementary material provides the implementation, trained allocator checkpoints, data splits and configurations, and scripts required to reproduce the reported experiments and analyses. The paper and supplementary material document the data preparation, codec checkpoints, allocation and selection procedures, bitstream accounting, evaluation protocols, and statistical analyses. We also provide the instructions and paired audio used for subjective evaluation. Third-party resources are redistributed only where permitted by their licenses; otherwise, we provide identifiers and preparation instructions. The code and configurations will be released publicly upon publication.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=33XGfHLtZg.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then Test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *The Sixth International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1802.01436.

Stephen Bates, Anastasios N. Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):43:1–43:34, 2021. doi: 10.1145/3478535.

Dimitrios Bralios, Paris Smaragdis, and Minje Kim. Elastic time: Dynamic frame rate bottlenecks for neural audio coding. In *Proc. Interspeech 2026*, 2026. To appear.

Karlheinz Brandenburg and Gerhard Stoll. ISO/MPEG-1 audio: A generic standard for coding of high-quality digital audio. *Journal of the Audio Engineering Society*, 42(10):780–792, October 1994.

Yunkee Chae and Kyogu Lee. Towards bitrate-efficient and noise-robust speech coding with variable bitrate RVQ. In *Proc. Interspeech 2025*, pp. 609–613, 2025. doi: 10.21437/Interspeech.2025-1253.

Yunkee Chae, Woosung Choi, Yuhta Takida, Junghyun Koo, Yukara Ikemiya, Zhi Zhong, Kin Wai Cheuk, Marco A. Martínez-Ramírez, Kyogu Lee, Wei-Hsiang Liao, and Yuki Mitsufuji. Variable bitrate residual vector quantization for audio coding. In *IEEE International Conference on*

Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10889508.

Weidong Chen, Helen M. Meng, and Xixin Wu. Gogo: Group-wise granularity-ordered codec for stable and efficient speech generation. In *International Conference on Learning Representations*, 2026.

Michael Chinen, Felicia S. C. Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines. ViSQOL v3: An open source production ready objective speech and audio metric. In *Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, 2020. doi: 10.1109/QoMEX48832.2020.9123150.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

Cisco Open Source. LRAC Challenge 2025: Data preparation tools. GitHub repository, archived 2025 recipe, 2026. URL https://github.com/cisco-open/lrac_data_generation/tree/lrac-2025-final. Release: lrac-2025-final.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jesper Jensen and Cees H. Taal. An algorithm for predicting the intelligibility of speech masked by modulated noise maskers. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2009–2022, 2016. doi: 10.1109/TASLP.2016.2585878.

J. D. Johnston. Transform coding of audio signals using perceptual noise criteria. *IEEE Journal on Selected Areas in Communications*, 6(2):314–323, 1988. doi: 10.1109/49.608.

Biing-Hwang Juang and A. H. Gray. Multiple stage vector quantization for speech coding. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 7, pp. 597–600, 1982. doi: 10.1109/ICASSP.1982.1171604.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. In *Advances in Neural Information Processing Systems*, volume 36, pp. 27980–27993, 2023.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. SDR—half-baked or well done? In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 626–630, 2019. doi: 10.1109/ICASSP.2019.8683855.

Jiaqi Li, Yao Qian, Yuxuan Hu, Leying Zhang, Xiaofei Wang, Heng Lu, Manthan Thakker, Jinyu Li, Sheng Zhao, and Zhizheng Wu. FlexiCodec: A dynamic neural audio codec for low frame rates. In *International Conference on Learning Representations*, 2026.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087, 2020.

Harikrishna Narasimhan, Wittawat Jitkrittum, Aditya K. Menon, Ankit Rawat, and Sanjiv Kumar. Post-hoc estimators for learning to defer to an expert. In *Advances in Neural Information Processing Systems*, volume 35, pp. 29292–29304, 2022.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

S. A. Ramprashad. High quality embedded wideband speech coding using an inherently layered coding paradigm. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pp. 1145–1148, 2000. doi: 10.1109/ICASSP.2000.859167.

A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra. Perceptual evaluation of speech quality (PESQ)—a new method for speech quality assessment of telephone networks and codecs. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pp. 749–752, 2001. doi: 10.1109/ICASSP.2001.941023.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *Proc. Interspeech 2022*, pp. 4521–4525, 2022. doi: 10.21437/Interspeech.2022-439.

Yair Shoham and Allen Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9):1445–1453, September 1988. doi: 10.1109/29.90373.

Hoyeol Sohn and Juhan Nam. DTM-Codec: Dynamic token masking for VFR speech coding with efficient boundary selection. In *Proc. Interspeech 2026*, 2026. To appear.

Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2125–2136, 2011. doi: 10.1109/TASL.2011.2114881.

Zixiang Wan, Haoran Zhao, Guochang Zhang, Runqiang Han, Jianqiang Wei, and Yuexian Zou. PhoenixCodec: Taming neural speech coding for extreme low-resource scenarios. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026. doi: 10.1109/ICASSP55912.2026.11462082.

Hankun Wang, Yiwei Guo, Chongtian Shao, Bohan Li, and Kai Yu. CodecSlime: Temporal redundancy compression of neural speech codec via dynamic frame rate. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 17017–17021, 2026. doi: 10.1109/ICASSP55912.2026.11463766.

Yunpeng Xu, Wenge Guo, and Zhi Wei. Selective conformal risk control. *arXiv preprint arXiv:2512.12844*, 2025. URL https://arxiv.org/abs/2512.12844.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2022. doi: 10.1109/TASLP.2021.3129994.

Hanglei Zhang, Yiwei Guo, Zhihan Li, Xiang Hao, Xie Chen, and Kai Yu. Unlocking temporal flexibility: Neural speech codec with variable frame rate. In *Proc. Interspeech 2025*, pp. 5003–5007, 2025. doi: 10.21437/Interspeech.2025-1289.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. SpeechTokenizer: Unified speech tokenizer for speech large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.16692.

Mingyu Zhao and Zhiyong Wu. Unistream: Multi-expert residual vector quantization for 48 khz causal streaming audio coding. *arXiv preprint arXiv:2609.09866*, 2026.

Mingyu Zhao, Zijian Lin, Yutang Feng, Jiatao Chen, Fan Wang, Jiehui Luo, Yuhao Ding, Jinchao Zhang, and Zhiyong Wu. BAMU: Bitstream-aware marginal-utility allocation for frozen pretrained neural speech codecs. *arXiv preprint arXiv:2608.08432*, 2026.

Rui-Chen Zheng, Wenrui Liu, Hui-Peng Du, Qinglin Zhang, Chong Deng, Qian Chen, Wen Wang, Yang Ai, and Zhen-Hua Ling. Say more with less: Variable-frame-rate speech tokenization via adaptive clustering and implicit duration coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35021–35029, 2026. doi: 10.1609/aaai.v40i41.40807.

A METHOD DETAILS

A.1 REFERENCE POLICIES AND ALLOCATOR ARCHITECTURE

Ref is a frozen codec-specific allocation policy, not necessarily uniform allocation. DAC, SpeechTokenizer, and PhoenixCodec use frozen reference cost predictors. Writing their predicted depth cost as $C_{t,d}$, the cumulative score is $v^{\mathrm{ref}}_{t,d} = -C_{t,d}$ and the marginal score for each additional stage is

$$u^{\mathrm{ref}}_{t,k} = v^{\mathrm{ref}}_{t,k} - v^{\mathrm{ref}}_{t,k-1}, \qquad k = 2, \ldots, K. \tag{15}$$

These scores anchor the learned correction. The reference allocation is obtained with its own fixed policy; a zero learned correction is not, by itself, a claim that proposal and reference solvers have identical penalties.

The allocator normalizes the first-stage latent using channel statistics estimated on fitting data and stored with the checkpoint. A pointwise convolution maps it to 128 channels. The normalized operating point $(\bar{d}-1)/(K-1)$ passes through a two-layer MLP with a SiLU activation, producing a feature-wise scale and shift. The scale is applied as $1 + 0.1\tanh(s)$. Four residual temporal blocks use dilations $1, 2, 4, 8$. Each block contains eight-group normalization, SiLU, a kernel-five depthwise convolution, a pointwise value/gate projection, sigmoid gating, and a pointwise output projection. Separate pointwise heads predict $K-1$ corrections for each supported operating point. Head weights and biases are initialized to zero.

Let $a_{t,k}$ be a head output and $s_u = \max\{\mathrm{std}_{t,k}(u^{\mathrm{ref}}_{t,k}), 10^{-4}\}$, using the population standard deviation over additional-stage marginals. The bounded correction is

$$\Delta u_{t,k} = 0.35 s_u \tanh(a_{t,k}). \tag{16}$$

Three independently seeded allocators produce utility tables that are averaged before optimization. They are ensemble members, not independent repetitions of a three-model experiment. The current proposal does not use the member-disagreement rejection rule present in earlier development code.

A.2 TRAINING AND BUDGET-CONSTRAINED OPTIMIZATION

Training uses a frozen bank of 25 complete candidate allocations per input condition and operating point, scored by PESQ against clean targets. The bank provides allocation-level supervision; it does not identify true frame-wise marginal gains. Candidate paths and their scores remain fixed during allocator fitting. The bank contains the reference-controller and first-stage-energy paths, three constant mixtures (weights 0.25, 0.50, and 0.75), sixteen single-feature sigmoid mixtures (energy, temporal motion, local roughness, or stage index, with coefficients $-2, -1, 1, 2$), and four joint energy/stage mixtures (coefficients ± 1). Energy and reference marginals are separately RMS-normalized before mixing; feature standardization is clipped to $[-4, 4]$. Each mixed score table is converted to an exact-budget path.

For candidate i, let s_i be its additive predicted score and Q_i its PESQ score. Within an input, predictions and targets are standardized as

$$z_i = \frac{s_i - \overline{s}}{\max\{\mathrm{std}(s), 10^{-4}\}}, \qquad \widetilde{Q}_i = \frac{Q_i - \overline{Q}}{\max\{\mathrm{std}(Q), 0.005\}}. \tag{17}$$

The listwise target is $p_i = \mathrm{softmax}(Q/0.02)_i$ and the prediction is $\widehat{p}_i = \mathrm{softmax}(z)_i$. The loss is

$$\mathcal{L} = -\sum_i p_i \log \widehat{p}_i + 0.25\,\mathrm{SmoothL1}(z, \widetilde{Q}) + 0.02\mathcal{L}_{\mathrm{dim}} + 0.01\mathcal{L}_{\mathrm{nonneg}}. \tag{18}$$

The last two terms penalize positive differences $u_{t,k+1} - u_{t,k}$ and negative marginals, respectively, using the mean rectified value divided by $\max\{\mathrm{mean}\,|u|, 10^{-4}\}$. The denominator is detached from the gradient. These are soft penalties, not hard monotonicity constraints. The temporal switching penalty is used in allocation, not in the supervised candidate path score.

For exact-budget inference, define $V_t(d) = \sum_{k=2}^{d} \bar{u}_{t,k}$, with $V_t(1) = 0$. Let $F_t(b, d)$ be the best score for the first t frames, total depth b, and current depth d. The dynamic program is

$$F_t(b, d) = V_t(d) + \max_{d' \in \{1,\ldots,K\}} \{F_{t-1}(b - d, d') - \lambda |d - d'|\}. \tag{19}$$

Initialize $F_1(d, d) = V_1(d)$ and all infeasible states to $-\infty$. The final value is $\max_d F_T(B, d)$; traceback recovers the allocation. The same fixed tie order must be used at both ends. The dense implementation takes $O(TBK^2)$ time, $O(BK)$ rolling score memory, and $O(TBK)$ traceback storage. Since $B = T\bar{d}$, this is not a constant-cost per-frame procedure. All reported main proposals use $\lambda = 0.1$.

A.3 Bitstream and Deterministic Coordination

The packet header specifies the codec operating configuration, frame count, and utterance-level mode. All mandatory first-stage indices precede the later-stage payload. The receiver first reconstructs the canonical first-stage latent, computes the selected allocation, then parses $d_t - 1$ later-stage indices for each frame. The mode communicates the encoder's selection; the receiver does not rerun fidelity or UTMOS evaluation. Shared-prefix allocation removes the frame-wise map, not the header or byte-alignment overhead. The binary choice has one bit of semantic information but need not add one separately packed byte or bit to the serialized packet.

Agreement requires identical model parameters, latent reconstruction, normalization, arithmetic conventions, and solver tie-breaking. Tests of a particular implementation do not establish universal cross-platform bit-exactness. Canonical-prefix, recovered-allocation, packet-size, and reconstruction checks should be interpreted within their recorded numerical configuration. The recovered DAC comparison disables TF32 consistently across arms after an earlier canonical-prefix check failed; it does not mix partial outputs from the earlier numerical configuration.

B Selective Deployment and Risk Evaluation

B.1 Selection Criteria and Threshold Selection

The fidelity criterion compares multi-resolution Mel losses against the encoder input, not the unavailable clean target. Let L_{ref} and L_{prop} be these losses. Its normalized score is $(L_{\mathrm{ref}} - L_{\mathrm{prop}})/s_{\mathrm{fid}}$, where the positive scale is frozen per codec and operating point. The archived development procedure uses the median absolute raw gain with a 10^{-8} floor. The perceptual criterion is the proposal-minus-reference UTMOS score, and requires a strictly positive difference.

The fidelity-only development threshold is chosen from the grid $\{0, 0.05, 0.1, 0.2, 0.3, 0.5, 0.75, 1, 1.5, 2\}$. The archived screening procedure requires acceptance of at least 10%, an all-input mean PESQ gain of at least 0.001, and an empirical cluster event fraction of at most 5%. Among feasible thresholds it maximizes acceptance, breaking ties in favor of the smaller threshold; infeasible settings fall back to Ref. This empirical screening does not supply a finite-sample risk guarantee. In particular, its development event-rate target must not be confused with a guaranteed confirmation risk. The table gives the deployed scales and thresholds read from policy artifacts. The UTMOS threshold is zero for all settings. Historical result identifiers ending in "tau0" refer to this perceptual threshold, not to zero fidelity threshold. The perceptual gate is subsequently combined with this frozen fidelity rule; confirmation outcomes are not used to retune either threshold.

Table 5: Frozen fidelity scales and thresholds; the additional UTMOS threshold is zero.

Codec	$\bar{d}$	Scale	τ
DAC	2	0.01673162	0.2
DAC	4	0.00518614	1.0
DAC	6	0.00447011	0.5
SpeechTokenizer	2	0.02207386	0.3
SpeechTokenizer	4	0.00992680	0.05
SpeechTokenizer	6	0.00592482	0.75
PhoenixCodec	2	0.00905198	1.0
PhoenixCodec	3	0.00993145	1.0
PhoenixCodec	4	0.00833642	1.5
PhoenixCodec	5	0.00638825	0.2

Fidelity-only and perceptual-only controls apply the corresponding criterion to the same proposal/reference outputs. The conjunction accepts their intersection. Consequently, adding a criterion cannot increase the unconditional event count on the same evaluation inputs; a lower event count alone is not evidence of better ranking at fixed acceptance.

B.2 Cluster Construction and Risk Estimation

Each utterance cluster contains clean, noisy, and reverberant conditions. Selection is applied to each input independently. A cluster incurs an event when at least one accepted proposal has PESQ more than 0.05 below its same-budget reference, with both outputs scored against the same clean target. The denominator is all 1,024 clusters, not accepted inputs or accepted clusters. Acceptance and deployed mean gains instead average over all 3,072 inputs, including zero gains for reference fallbacks.

We recompute the main table's one-sided pointwise 95% Clopper–Pearson bounds from the archived event counts, using $\delta = 0.05$ for each setting. They are not the multiplicity-adjusted bounds stored for earlier, larger selector-comparison families. Table 6 gives the complete counts and pointwise bounds under the current reporting convention.

Table 6: Confirmation on 1,024 clusters per setting. P/S denote Proposal-Only/Selective; gain and acceptance describe Selective; U is its pointwise 95% upper bound.

Codec	$\bar{d}$	P events	S events	Gain	Accept (%)	U (%)
DAC	2	363	38	0.0077	14.58	4.84
DAC	4	272	25	0.0040	12.76	3.39
DAC	6	280	41	0.0045	15.23	5.16
SpeechTokenizer	2	248	24	0.0101	17.64	3.28
SpeechTokenizer	4	300	30	0.0054	14.23	3.95
SpeechTokenizer	6	256	21	0.0029	9.18	2.94
PhoenixCodec	2	328	31	0.0163	13.22	4.06
PhoenixCodec	3	392	32	0.0202	13.41	4.17
PhoenixCodec	4	385	33	0.0141	9.80	4.29
PhoenixCodec	5	269	39	0.0141	21.84	4.95

Coverage assumes independent identically distributed clusters and a policy and event definition fixed independently of their evaluation outcomes. Source exclusion checks do not prove speaker independence or eliminate unknown overlap with pretrained codecs. The results do not imply simultaneous coverage across settings, conditional-on-acceptance safety, subjective preference guarantees, or robustness to a different deployment distribution.

C Experimental Setup and Reproducibility

C.1 Datasets, Splits, and Overlap Checks

LRAC data construction supplies source speech, noise, and room responses. The public Track-1 set contains 600 clean, 200 noisy, and 200 reverberant inputs. The controlled allocator bank uses 640 source identities for fitting and 128 for checkpoint selection, keeping conditions of the same source together. These counts describe fitting and model selection, not the confirmation population.

The risk-evaluation reservation contains calibration and confirmation cohorts of 1,024 clusters each, with three acoustic conditions per cluster. The documented construction excludes known allocator/selector inputs and the Phoenix training manifest, applies content-identity checks, and orders eligible source contents deterministically before splitting. Such reservation separates known project inputs but does not prove population-random sampling, speaker disjointness, or exclusion from external codec pretraining. SpeechTokenizer uses the reserved cohorts after other codec experiments; they should not be described as globally unseen throughout the research project. Previously accessed calibration cohorts remain descriptive when reused for ablations.

The separate fixed test-clean subset contains 1,000 utterances and 40 speakers, with 202 utterances shorter than three seconds, 492 between three and six seconds, and 306 between six and ten sec-

onds. Source metadata identify overlapping LibriSpeech chapters with the development source pool. It therefore supports descriptive reconstruction and selection comparisons, not independent confirmation of a newly selected policy.

C.2 Codec Configurations and Training Settings

For Proposal, the encoder, decoder, and RVQ codebooks remain frozen. DAC and SpeechTokenizer use eight RVQ stages and mean depths 2, 4, and 6; PhoenixCodec uses six stages and mean depths 2, 3, 4, and 5. SpeechTokenizer runs at 16 kHz; the other codecs run at 24 kHz. SpeechTokenizer outputs are resampled to the common evaluation rate using the same preprocessing for all compared methods.

The structured allocator recipe uses AdamW with learning rate 3×10^{-4}, weight decay 10^{-4}, batch size 32, and gradient-norm clipping at 1.0 for 10,000 updates. Batches sample candidate-bank rows with replacement. Checkpoints are evaluated every 250 updates, selecting the one with minimum candidate-bank PESQ regret on the selection partition. There is no learning-rate schedule in this training loop. Three models are trained per codec; normalization statistics and supported rate heads are saved with each model. Reference-controller fitting and structured-allocator fitting are separate stages. The reference cost predictor is fixed before structured-allocator training and is not updated during that stage. The 10,000-update budget reported above applies only to the structured allocator and excludes reference-controller fitting.

C.3 Baselines and Budget Matching

CBR uses a constant RVQ depth at the requested operating point. Ref uses the codec-specific frozen reference policy. Proposal uses learned shared-prefix allocation without deployment selection; Selective chooses between Proposal and Ref. Allocation-quality comparisons match $B = T\bar{d}$ code indices. This equality does not by itself equate bitrates across different codecs.

Coordination comparisons instead charge actual serialized bytes, including the header, RVQ indices, allocation signaling, and padding. Compressed-Map uses the same utility predictor but sends the depth sequence, reducing the budget available for indices. The map syntax uses run lengths capped at 16, four bits for run length minus one, and $\lceil \log_2 K \rceil$ bits for depth. Explicit-Map stores uncompressed per-frame counts. The compressed-map allocation implementation uses a Lagrangian sequence solver with repair/fill rather than the same exact index-budget DP. Thus the operational comparison includes the charged-map optimization procedure as well as the signaling cost; it does not isolate a purely byte-removal intervention on an unchanged allocation.

C.4 Metrics and Statistical Procedures

We report PESQ (Rix et al., 2001), UTMOS (Saeki et al., 2022), ViSQOL (Chinen et al., 2020), STOI (Taal et al., 2011), ESTOI (Jensen & Taal, 2016), and SI-SDR (Le Roux et al., 2019). Reference-based scores compare each reconstruction with the clean target; the selection-time Mel criterion instead uses the actual encoder input. UTMOS is reference-free. PESQ and UTMOS are aligned with training and selection, respectively, and should not be treated as independent confirmation of the perceptual objectives they helped optimize.

Test-clean intervals use paired speaker-bootstrap resampling, with 10,000 replicates and percentile endpoints. Controlled-condition intervals resample whole source clusters, preserving their paired views and method outputs. These pointwise intervals do not provide simultaneous inference across all metrics and operating points. Invalid scoring cases must use the same exclusion cohort for all compared variants, as in the allocator ablation below.

Random@Acceptance samples the same number of accepted proposals as Selective without replacement within each codec–operating-point cell and averages 1,000 draws. This is a retrospective matched-usage diagnostic, not a deployable policy fitted independently of the batch. Variability across random draws is distinct from sampling uncertainty across utterances or speakers.

D Extended Results and Ablations

D.1 Full Fixed-Budget Results

Table 7 reports Proposal-minus-Ref contrasts for all six metrics and all operating points on test-clean. The allocator is ungated in these comparisons. These entries must not be confused with Selective-minus-Ref gains in the deployment experiment.

Table 7: Proposal-minus-Ref contrasts on test-clean, with pointwise 95% paired speaker-bootstrap intervals.

Codec	$\bar{d}$	Metric	Gain	95% interval
DAC	2	PESQ	+0.07203	[+0.06396, +0.08042]
DAC	2	UTMOS	+0.15893	[+0.14200, +0.17512]
DAC	2	VISQOL	+0.05077	[+0.03616, +0.06453]
DAC	2	STOI	+0.00859	[+0.00755, +0.00960]
DAC	2	ESTOI	+0.01230	[+0.01078, +0.01373]
DAC	2	SI-SDR	+0.03370	[-0.20459, +0.23531]
DAC	4	PESQ	+0.03396	[+0.02834, +0.04045]
DAC	4	UTMOS	+0.05068	[+0.04482, +0.05663]
DAC	4	VISQOL	+0.02166	[+0.01274, +0.03039]
DAC	4	STOI	+0.00247	[+0.00214, +0.00280]
DAC	4	ESTOI	+0.00369	[+0.00319, +0.00426]
DAC	4	SI-SDR	-0.02065	[-0.10176, +0.07152]
DAC	6	PESQ	+0.05281	[+0.03937, +0.06677]
DAC	6	UTMOS	+0.06662	[+0.05582, +0.07818]
DAC	6	VISQOL	+0.00760	[-0.00633, +0.02144]
DAC	6	STOI	+0.00282	[+0.00187, +0.00371]
DAC	6	ESTOI	+0.00462	[+0.00311, +0.00607]
DAC	6	SI-SDR	+0.02100	[-0.13521, +0.20737]
SpeechTokenizer	2	PESQ	+0.02850	[+0.02447, +0.03248]
SpeechTokenizer	2	UTMOS	+0.10620	[+0.09402, +0.11838]
SpeechTokenizer	2	VISQOL	+0.03493	[+0.02627, +0.04428]
SpeechTokenizer	2	STOI	+0.00747	[+0.00653, +0.00846]
SpeechTokenizer	2	ESTOI	+0.01062	[+0.00924, +0.01193]
SpeechTokenizer	2	SI-SDR	+0.55164	[+0.49777, +0.60530]
SpeechTokenizer	4	PESQ	+0.02345	[+0.01854, +0.02844]
SpeechTokenizer	4	UTMOS	+0.03580	[+0.02863, +0.04225]
SpeechTokenizer	4	VISQOL	+0.04515	[+0.03334, +0.05703]
SpeechTokenizer	4	STOI	+0.00311	[+0.00275, +0.00348]
SpeechTokenizer	4	ESTOI	+0.00439	[+0.00374, +0.00504]
SpeechTokenizer	4	SI-SDR	+0.24713	[+0.21815, +0.27880]
SpeechTokenizer	6	PESQ	+0.00587	[-0.00084, +0.01266]
SpeechTokenizer	6	UTMOS	+0.00661	[+0.00162, +0.01159]
SpeechTokenizer	6	VISQOL	+0.00367	[-0.01719, +0.02531]
SpeechTokenizer	6	STOI	+0.00211	[+0.00175, +0.00244]
SpeechTokenizer	6	ESTOI	+0.00284	[+0.00215, +0.00343]
SpeechTokenizer	6	SI-SDR	+0.29524	[+0.25474, +0.32927]
PhoenixCodec	2	PESQ	+0.04344	[+0.03667, +0.05002]
PhoenixCodec	2	UTMOS	+0.04311	[+0.03298, +0.05444]
PhoenixCodec	2	VISQOL	+0.01806	[+0.00890, +0.02753]
PhoenixCodec	2	STOI	+0.00179	[+0.00132, +0.00230]
PhoenixCodec	2	ESTOI	+0.00247	[+0.00181, +0.00315]
PhoenixCodec	2	SI-SDR	+0.03531	[-0.13650, +0.20823]
PhoenixCodec	3	PESQ	+0.03878	[+0.03173, +0.04580]
PhoenixCodec	3	UTMOS	+0.01811	[+0.00933, +0.02725]
PhoenixCodec	3	VISQOL	-0.00880	[-0.01651, -0.00106]
PhoenixCodec	3	STOI	+0.00097	[+0.00057, +0.00134]
PhoenixCodec	3	ESTOI	+0.00131	[+0.00069, +0.00193]
PhoenixCodec	3	SI-SDR	-0.03960	[-0.19463, +0.10858]
PhoenixCodec	4	PESQ	+0.02065	[+0.01549, +0.02639]
PhoenixCodec	4	UTMOS	+0.00597	[+0.00041, +0.01123]
PhoenixCodec	4	VISQOL	-0.00738	[-0.01889, +0.00325]
PhoenixCodec	4	STOI	-0.00003	[-0.00039, +0.00029]

Codec	$\bar{d}$	Metric	Gain	95% interval
PhoenixCodec	4	ESTOI	+0.00024	[-0.00030, +0.00072]
PhoenixCodec	4	SI-SDR	-0.06250	[-0.14594, +0.02102]
PhoenixCodec	5	PESQ	+0.00657	[+0.00380, +0.00943]
PhoenixCodec	5	UTMOS	-0.00010	[-0.00251, +0.00224]
PhoenixCodec	5	VISQOL	-0.00983	[-0.02114, +0.00091]
PhoenixCodec	5	STOI	+0.00039	[+0.00026, +0.00052]
PhoenixCodec	5	ESTOI	+0.00056	[+0.00034, +0.00076]
PhoenixCodec	5	SI-SDR	+0.01643	[-0.02941, +0.06704]

D.2 Rate–Distortion Curves

Figure 4 complements the fixed-budget contrasts with absolute PESQ and ViSQOL scores across measured rates on the same 1,000-utterance LibriSpeech test-clean cohort. We evaluate CBR, Ref, Proposal, and Selective at $\bar{d} \in \{2, 4, 6\}$ for DAC and SpeechTokenizer, and $\bar{d} \in \{2, 3, 4, 5\}$ for PhoenixCodec. Selective uses the frozen deployment policy described in Appendix B.1.

For each codec, operating point, and method, the total serialized rate is

$$R_{\text{total}} = \frac{8 \sum_{i=1}^{N} S_i}{1000 \sum_{i=1}^{N} L_i} \quad \text{kbps}, \tag{20}$$

where S_i is the complete serialized packet size in bytes and L_i is the input duration in seconds. This accounting includes the 14-byte header, RVQ payload, and terminal byte-alignment padding. Rate is computed from aggregate bytes and duration, whereas quality scores are averaged over utterances. Within each codec and operating point, all four methods have identical per-utterance serialized packet sizes. Codec-specific frame rates and codebook configurations mean that the same $\bar{d}$ need not yield the same bitrate across codecs.

D.3 Coordination Quality and Timing Results

Table 8 reports 1,000 LRAC public-test inputs at $\bar{d} = 4$ and matched total transmitted bytes. Time is median receiver coordination time, excluding waveform decoding, not end-to-end latency. Shared-Prefix has higher mean ViSQOL than Compressed-Map for all three codecs; both remain below Ref on PhoenixCodec. The comparison includes solver differences as well as signaling overhead.

Table 8: Complete coordination results. Gains are ViSQOL relative to Ref; map size is mean bytes per input and time is median milliseconds.

Codec	Method	Gain	Map (B)	Time (ms)
DAC	Shared-Prefix	0.019	0.0	508.88
DAC	Compressed-Map	0.007	35.2	0.67
SpeechTokenizer	Shared-Prefix	0.028	0.0	316.54
SpeechTokenizer	Compressed-Map	0.014	24.6	0.43
PhoenixCodec	Shared-Prefix	-0.003	0.0	462.50
PhoenixCodec	Compressed-Map	-0.010	33.0	0.76

D.4 Selector Controls and Threshold Sweeps

Table 9 extends the main selector ablation to all operating points. The event rate here is input-level on the descriptive test-clean set, not the three-condition cluster-level confirmation risk. No threshold is selected from these reported test-clean outcomes.

Table 9: Test-clean selector controls. Gain averages over all inputs; Drop is input-level.

Codec	$\bar{d}$	Selector	Accept (%)	PESQ gain	Drop (%)
DAC	2	Proposal	100.0	+0.07203	2.20

Codec	$\bar{d}$	Selector	Accept (%)	PESQ gain	Drop (%)
DAC	2	Fidelity	20.8	+0.02176	0.10
DAC	2	Perceptual	91.6	+0.06855	1.60
DAC	2	Selective	19.6	+0.02097	0.10
DAC	4	Proposal	100.0	+0.03396	2.30
DAC	4	Fidelity	17.4	+0.01138	0.20
DAC	4	Perceptual	81.0	+0.02895	1.60
DAC	4	Selective	14.9	+0.00961	0.20
DAC	6	Proposal	100.0	+0.05281	8.60
DAC	6	Fidelity	16.2	+0.01539	0.40
DAC	6	Perceptual	83.9	+0.04942	6.30
DAC	6	Selective	14.7	+0.01432	0.40
SpeechTokenizer	2	Proposal	100.0	+0.02850	1.90
SpeechTokenizer	2	Fidelity	28.0	+0.01239	0.30
SpeechTokenizer	2	Perceptual	79.9	+0.02490	1.20
SpeechTokenizer	2	Selective	23.7	+0.01121	0.10
SpeechTokenizer	4	Proposal	100.0	+0.02345	3.10
SpeechTokenizer	4	Fidelity	29.4	+0.01216	0.20
SpeechTokenizer	4	Perceptual	65.1	+0.02040	1.70
SpeechTokenizer	4	Selective	21.1	+0.00948	0.20
SpeechTokenizer	6	Proposal	100.0	+0.00587	11.00
SpeechTokenizer	6	Fidelity	9.7	+0.00366	0.00
SpeechTokenizer	6	Perceptual	50.4	+0.00585	4.90
SpeechTokenizer	6	Selective	5.5	+0.00247	0.00
PhoenixCodec	2	Proposal	100.0	+0.04344	7.60
PhoenixCodec	2	Fidelity	22.6	+0.02019	0.90
PhoenixCodec	2	Perceptual	64.1	+0.03853	2.60
PhoenixCodec	2	Selective	17.3	+0.01765	0.70
PhoenixCodec	3	Proposal	100.0	+0.03878	10.30
PhoenixCodec	3	Fidelity	11.9	+0.01228	0.80
PhoenixCodec	3	Perceptual	54.2	+0.03200	3.50
PhoenixCodec	3	Selective	8.3	+0.00960	0.40
PhoenixCodec	4	Proposal	100.0	+0.02065	8.20
PhoenixCodec	4	Fidelity	3.6	+0.00263	0.30
PhoenixCodec	4	Perceptual	52.7	+0.01465	3.40
PhoenixCodec	4	Selective	2.3	+0.00193	0.00
PhoenixCodec	5	Proposal	100.0	+0.00657	3.90
PhoenixCodec	5	Fidelity	26.3	+0.00415	0.00
PhoenixCodec	5	Perceptual	51.1	+0.00470	1.60
PhoenixCodec	5	Selective	14.5	+0.00301	0.00

Table 10: Matched-acceptance controls. S/R denote Selective/Random; random values average 1,000 draws.

Codec	$\bar{d}$	Count	S gain	R gain	S drop (%)	R drop (%)
DAC	2	196	+0.02097	+0.01412	0.10	0.43
DAC	4	149	+0.00961	+0.00504	0.20	0.34
DAC	6	147	+0.01432	+0.00776	0.40	1.25
SpeechTokenizer	2	237	+0.01121	+0.00676	0.10	0.45
SpeechTokenizer	4	211	+0.00948	+0.00493	0.20	0.66
SpeechTokenizer	6	55	+0.00247	+0.00032	0.00	0.60
PhoenixCodec	2	173	+0.01765	+0.00755	0.70	1.31
PhoenixCodec	3	83	+0.00960	+0.00323	0.40	0.85
PhoenixCodec	4	23	+0.00193	+0.00048	0.00	0.19
PhoenixCodec	5	145	+0.00301	+0.00095	0.00	0.56

Calibration curves sweep the normalized fidelity threshold while fixing positive UTMOS gain. Their accepted sets are nested. The frozen operating points are read from the deployment policies, not selected to optimize the plotted curves. Table 11 records the marked points at the lowest common operating point, $\bar{d} = 2$.

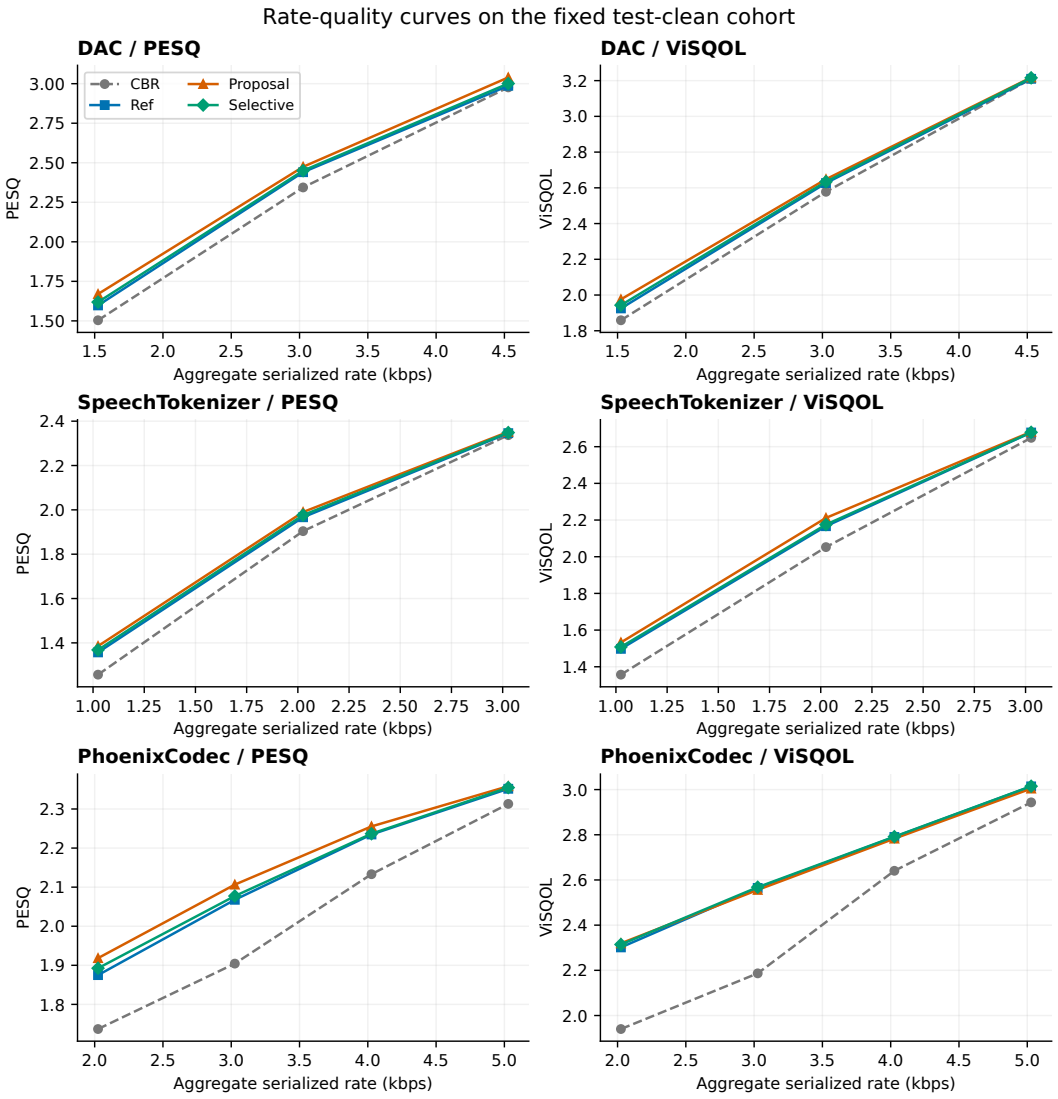

Figure 4: Rate–distortion curves on the fixed 1,000-utterance test-clean cohort. Rows show DAC, SpeechTokenizer, and PhoenixCodec; columns show PESQ and ViSQOL (higher is better). The horizontal axis reports total serialized rate, including the header and byte-alignment padding. CBR, Ref, Proposal, and Selective use matched per-utterance byte budgets within each codec and operating point. Lines connect evaluated operating points; intermediate bitrates were not evaluated.

Table 11: Frozen calibration points at $\bar{d} = 2$ (1,024 clusters, 3,072 inputs per codec).

Codec	τ	Accepted	Events	Accept (%)	Drop (%)
DAC	0.2	447	35	14.55	3.42
SpeechTokenizer	0.3	548	17	17.84	1.66
PhoenixCodec	1.0	389	31	12.66	3.03

D.5 Allocator Ablations

We hold the learned checkpoints and index budget fixed at $\bar{d} = 4$, disable selection, and vary ensemble membership or the solver's smoothness penalty. Removing smoothing requires no retraining. The common cohort has 1,021 clusters: three of 1,024 clean targets could not be scored against themselves by PESQ, so all their conditions are excluded for every variant. These are descriptive calibration-data ablations, not confirmation estimates.

Table 12: DAC allocator ablations on the common calibration cohort. Gain is mean PESQ relative to Ref; Drop is the cluster percentage with a PESQ decrease exceeding 0.05.

Allocator	Gain	Drop (%)
Single model 1	0.0119	23.31
Single model 2	0.0085	27.03
Single model 3	0.0097	26.35
Ensemble	0.0105	25.27
Ensemble, no smoothing	0.0115	35.95

The paired no-smoothing-minus-ensemble PESQ interval includes zero. The ensemble does not outperform member 1. The reduction in observed drop frequency should not be interpreted as a significance claim about drop rates based on a PESQ-mean interval.

D.6 External Baselines and Adaptation Protocols

BAMU reproduction. The DAC comparison uses 640 fitting and 128 development source segments, three seeds (20260921–20260923), and 40 training epochs. It does not reproduce BAMU's original 100-hour training recipe. The predictor uses convolution widths 128/128/64, kernel sizes 5/3/3, eight-group normalization, GELU, and eight marginal outputs. These normalization choices are local implementation decisions. Labels derive from native latent residual reduction rather than the PESQ-supervised targets of the shared-prefix allocator. Allocation uses four-frame blocks, 64 multiplier candidates, batched Viterbi and byte-checked greedy refinement. The input-scaled multiplier grid is a disclosed local choice.

All methods are evaluated on the same 600 clean, 200 noisy, and 200 reverberant public LRAC inputs. The research container uses a 24-byte header; explicit maps use three-bit depths and Elias-gamma run lengths, with map and code alignment charged separately. Map bytes reduce the available code payload. Unused byte slack and transport padding are distinct from useful payload. This research syntax is not asserted to be an official BAMU bitstream implementation. The DAC evaluation disables quantizer TF32 consistently after a native/canonical-prefix discrepancy was detected; the failed partial run is not merged with the corrected run. Raw allocation results do not measure selector performance. Receiver allocation computation remains necessary for the map-free arm; quality gains are not claims of latency superiority.

SP-Count. SP-Count is a Gogo-inspired shared-coarse count predictor with our frozen codec, candidate supervision, and evaluation pipeline, not a reproduction of the full Gogo model. Both raw proposals and common-gate reconstructions are scored on the public LRAC set. Its paired bootstrap groups 1,000 inputs into 949 clean-reference-hash clusters, not the three-condition clusters used for confirmation risk. At DAC depths 4 and 6, gated Proposal-minus-SP-Count PESQ differences are 0.00440 $[0.00306, 0.00581]$ and 0.01238 $[0.00916, 0.01606]$. At depth 2 the difference is -0.00091 $[-0.00238, 0.00060]$. These intervals are descriptive and pointwise; no population risk certificate is

Table 13: All archived FlexiCodec public-LRAC operating points. Rate includes the research transport header; scores are absolute means.

Setting	Total kbps	PESQ	ViSQOL
flexi_t086_q01	0.1215	1.0947	1.1993
flexi_t086_q04	0.3271	1.4650	1.9664
flexi_t086_q08	0.6017	1.8617	2.4343
flexi_t086_q12	0.8755	2.0675	2.6753
flexi_t086_q16	1.1501	2.1865	2.8095
flexi_t086_q24	1.6976	2.2648	2.8880
flexi_t091_q01	0.1592	1.1020	1.2073
flexi_t091_q04	0.4453	1.6472	2.1611
flexi_t091_q08	0.8271	2.0768	2.6849
flexi_t091_q12	1.2089	2.2604	2.8845
flexi_t091_q16	1.5896	2.3477	2.9743
flexi_t091_q24	2.3539	2.4026	3.0197
flexi_t100_q01	0.2091	1.1055	1.2272
flexi_t100_q04	0.6634	1.8901	2.4063
flexi_t100_q08	1.2694	2.3348	2.9643
flexi_t100_q12	1.8753	2.4648	3.0972
flexi_t100_q16	2.4813	2.5020	3.1331
flexi_t100_q24	3.6932	2.5207	3.1484

inferred. The independent comparison runs use their own recorded numerical settings and should not be merged row-wise with the BAMU table.

VRVQ adaptation and matched backbone. The archived experiment uses SonyResearch/VRVQ source commit `1033370e09b108a41b11fde31a959489dfe2d132`. Paired CBR and VRVQ endpoints are trained on 133,800 LRAC speech-training items for 300,000 updates at 24 kHz, batch size eight, and seed 20260823. Both use eight RVQ stages with 1,024 entries, encoder strides $[2, 4, 4, 10]$, and reversed decoder strides. This is an LRAC adaptation, not the original paper's 44.1-kHz training/evaluation setting.

The post-hoc controller freezes the CBR encoder, codebooks, and decoder. It learns rate-conditioned shared-prefix scores from low/mid/high multiscale spectral-distortion targets on 768 training-source utterances and their acoustic views. Base and residual stages use AdamW with learning rate 3×10^{-4}, batch size 32, and 5,000 updates each; checkpoint selection uses the training-cache holdout only. Its 381,472 parameters and supervision differ from the main structured ensemble. It is reported as a separate shared-prefix variant, not relabeled as the exact main Proposal.

CBR and post-hoc use identical file bytes, with a 24-byte header and ten bits per RVQ index. VRVQ's explicit three-bit depth map adds 225 bits/s and is included in measured rate. Contrasts interpolate VRVQ quality piecewise linearly at each input's matched actual bitrate; they are not fresh decodes at every interpolated point. Evaluation uses all 1,000 public LRAC inputs with 10,000 paired bootstrap draws. Public outcomes do not select checkpoints. VRVQ-minus-CBR PESQ differences are -0.1871, -0.0733, and -0.0040 at the three matched rates. Its weak lower-rate performance must be considered when interpreting post-hoc advantages; these results do not establish general superiority over end-to-end variable-rate training.

FlexiCodec system-level context. FlexiCodec (Li et al., 2026) is evaluated at native 16 kHz with a common 24-kHz scoring wrapper. Its separate public LRAC operating points charge codes, durations, and a 14-byte research transport header. The transport wrapper is not asserted to be the official wire format. Table 13 reports all 18 public-set points, not a quality-selected subset. This is system-level context, not a matched-backbone allocator ablation or a direct claim that our method dominates FlexiCodec. Same-quantizer changes in merging alter both rate and quality.

Table 14: DAC allocation comparisons on 1,000 public LRAC inputs. (a) Absolute scores at $\bar{d} = 4$ under the BAMU-style matched-total-byte protocol, without selection. (b) PESQ gains over Ref under a separate SP-Count protocol; raw proposals and common-gate outputs are reported separately. SP-Count is Gogo-inspired, not an official reproduction.

(a) Matched-data BAMU-style comparison

Method	PESQ	ViSQOL	Map (B)
CBR	2.0318	2.6144	0.0
Ref	2.1093	2.6501	0.0
BAMU-style, seed 1	2.0541	2.6421	79.5
BAMU-style, seed 2	2.0462	2.6410	80.5
BAMU-style, seed 3	2.0592	2.6463	80.0
Shared-Prefix	2.1379	2.6690	0.0

(b) SP-Count comparison

	Raw		Common gate	
$\bar{d}$	Proposal	SP-Count	Proposal	SP-Count
2	0.03456	0.01666	0.00391	0.00483
4	0.02839	-0.00216	0.00548	0.00108
6	0.04747	-0.00510	0.01128	-0.00110

Table 15: Matched-backbone LRAC comparison. PESQ differences favor the post-hoc controller. Rates include container costs; VRVQ is interpolated within its measured per-input rate range. Intervals are pointwise paired 95% bootstrap intervals.

Rate (kbps)	Post-hoc − CBR	Post-hoc − VRVQ	95% interval
1.5405	0.0558	0.2429	[0.2291, 0.2571]
3.0417	0.0015	0.0748	[0.0650, 0.0846]
4.5437	-0.0057	-0.0016	[-0.0094, 0.0057]

D.7 Encoder-LoRA Adaptation

This comparison tests sender-side parameter adaptation against allocation with a frozen codec. Only additive low-rank encoder updates are trained; original weights, biases, normalization, RVQ codebooks, and the decoder remain frozen. Each adapter has rank $r = 8$ and scaling parameter $\alpha = 8$, with Kaiming-initialized A and zero-initialized B. DAC targets encoder Conv1d kernels; SpeechTokenizer targets Conv1d and the input/recurrent matrices of its bidirectional LSTM; PhoenixCodec targets Conv2d and GRU matrices. Equal rank does not imply equal capacity.

Table 16: Encoder-LoRA adapter sizes from the recorded module identities.

Codec	Target entries	Trainable parameters
DAC	30	375,352
SpeechTokenizer	26	632,888
PhoenixCodec	18	50,040

Training uses 640 source speech items with clean/noisy/reverberant views, and a deterministic 96-view validation subset from the next 128 sources. Three runs use seeds 20261271, 20261272, and 20261273. Each run trains one adapter shared across rates for 8,000 AdamW updates, batch size eight, learning rate 2×10^{-4}, betas $(0.9, 0.95)$, no weight decay, and gradient clipping at one. Rates cycle through the codec's operating points. The loss combines multi-resolution STFT, waveform L1 (weight 2), and frozen-encoder latent MSE (weight 0.02). Checkpoints minimize validation reconstruction loss, evaluated every 400 updates; PESQ and UTMOS are not checkpoint-selection objectives. Quantizer forward values remain unchanged, with identity straight-through gradients. SpeechTokenizer preserves the fixed 24-to-16-to-24 kHz resampling path.

Evaluation uses the same 1,000 test-clean utterances and serialized-byte budgets as the named baseline. Runs are averaged, not ensembled at inference. Table 17 reports all operating points. Differences against Selective include the reference fallback; the Proposal contrast separates this from ungated allocation. The archived validation includes zero-update encoder parity, frozen-weight checks, and native decoder replay.

Table 17: Encoder-LoRA minus each baseline in mean PESQ; three-run averages. SD is across-run standard deviation, not a confidence interval.

Codec	$\bar{d}$	vs. CBR	vs. Proposal	vs. Selective	Seed SD
DAC	2	+0.0396	-0.1263	-0.0752	0.0059
DAC	4	+0.0018	-0.1287	-0.1043	0.0033
DAC	6	-0.1559	-0.2186	-0.1802	0.0079
SpeechTokenizer	2	+0.1423	+0.0135	+0.0307	0.0064
SpeechTokenizer	4	+0.1751	+0.0891	+0.1031	0.0084
SpeechTokenizer	6	+0.0907	+0.0757	+0.0791	0.0107
PhoenixCodec	2	-0.0525	-0.2339	-0.2081	0.0181
PhoenixCodec	3	-0.0071	-0.2092	-0.1801	0.0196
PhoenixCodec	4	-0.0646	-0.1869	-0.1682	0.0243
PhoenixCodec	5	-0.1190	-0.1643	-0.1607	0.0302

Table 18: LoRA-minus-baseline PESQ differences with pointwise 95% paired speaker-bootstrap intervals for the three-run average. These intervals do not resample training seeds.

Codec	$\bar{d}$	Baseline	Difference	95% interval
DAC	2	Proposal	-0.12630	[-0.14814, -0.10515]
DAC	2	Selective	-0.07524	[-0.09654, -0.05440]
DAC	4	Proposal	-0.12866	[-0.15431, -0.10373]
DAC	4	Selective	-0.10431	[-0.12865, -0.08054]
DAC	6	Proposal	-0.21864	[-0.25168, -0.18584]
DAC	6	Selective	-0.18015	[-0.20556, -0.15412]
SpeechTokenizer	2	Proposal	+0.01346	[-0.00424, +0.03227]
SpeechTokenizer	2	Selective	+0.03075	[+0.01269, +0.04936]
SpeechTokenizer	4	Proposal	+0.08909	[+0.06983, +0.11008]
SpeechTokenizer	4	Selective	+0.10306	[+0.08495, +0.12262]
SpeechTokenizer	6	Proposal	+0.07568	[+0.05590, +0.09725]
SpeechTokenizer	6	Selective	+0.07908	[+0.06110, +0.09932]
PhoenixCodec	2	Proposal	-0.23391	[-0.26175, -0.20638]
PhoenixCodec	2	Selective	-0.20811	[-0.23360, -0.18297]
PhoenixCodec	3	Proposal	-0.20924	[-0.23372, -0.18478]
PhoenixCodec	3	Selective	-0.18006	[-0.20407, -0.15593]
PhoenixCodec	4	Proposal	-0.18693	[-0.21470, -0.15985]
PhoenixCodec	4	Selective	-0.16821	[-0.19586, -0.14073]
PhoenixCodec	5	Proposal	-0.16431	[-0.19534, -0.13416]
PhoenixCodec	5	Selective	-0.16074	[-0.19122, -0.13124]

Table 19: Per-seed LoRA-minus-Selective PESQ differences; all three runs retained.

Codec	$\bar{d}$	20261271	20261272	20261273
DAC	2	-0.08188	-0.07081	-0.07305
DAC	4	-0.10561	-0.10051	-0.10682
DAC	6	-0.17183	-0.18764	-0.18099
SpeechTokenizer	2	+0.03464	+0.03427	+0.02334
SpeechTokenizer	4	+0.10586	+0.10969	+0.09363
SpeechTokenizer	6	+0.08043	+0.08904	+0.06778
PhoenixCodec	2	-0.19502	-0.22878	-0.20053
PhoenixCodec	3	-0.16162	-0.20072	-0.17784
PhoenixCodec	4	-0.14462	-0.19313	-0.16689

Codec	$\bar{d}$	20261271	20261272	20261273
PhoenixCodec	5	-0.13264	-0.19270	-0.15688

E Model Size and Computational Cost

E.1 Parameter Counts and Deployment Dependencies

The counts below describe the structured utility allocator only. They exclude the frozen codec, any separate frozen reference predictor, and encoder-side quality estimators. The three-model count includes all ensemble weights, not a single model reused three times. UTMOS is an encoder-side dependency even though its parameters are not trained in the proposed method; it is not required by the receiver.

Table 20: Structured allocator parameters from archived deployment receipts.

Codec	Single model	Three-model ensemble
DAC	369,429	1,108,287
SpeechTokenizer	369,429	1,108,287
PhoenixCodec	281,108	843,324

E.2 Profiling Protocol and Runtime Breakdown

The archived DAC component profile uses eight source utterances, each with clean, noisy, and reverberant views and two measured repeats (48 observations per codec–rate cell). The short inputs are 1.2 s. The recorded environment is an RTX 4090 workstation, not a mobile receiver. Two additional source inputs are used for warm-up. Initialization is excluded from steady-state measurements. The following table reports individual recorded component medians; their sum is not asserted to equal the median end-to-end deployment time.

Table 21: Archived DAC short-input component profiling at $\bar{d} = 4$, excluding UTMOS; not full current-policy end-to-end latency.

Component	Median (ms)
Encoder trace	15.485
Prefix reconstruction	0.174
Joint allocation scheduling	100.667
Paired waveform decoding	26.540
Mel comparison	5.840

These diagnostics differ in input length and timing boundary from the 1,000-input coordination experiment. They must not be substituted for its receiver medians. Historical combined selector timing estimates add separately measured components, including perceptual scoring; they are not native end-to-end timings of the full current deployment path. The reported profile does not establish three-codec end-to-end latency, peak deployment memory, energy consumption, or real-time streaming behavior. DP computation grows with utterance length and exact budget, whereas selection additionally requires proposal/reference reconstruction and quality scoring at the encoder.

F Subjective Evaluation

F.1 Study Protocol and Evaluation Materials

We report the DAC, SpeechTokenizer, and PhoenixCodec panels from a blinded, randomized MUSHRA-style listening study. Scores range from 0 to 100; panels include clean references, hidden references, and low-pass anchors. Candidate order is randomized and method identities are

hidden. Competing reconstructions are matched in total transmitted bits. DAC uses LRAC public clean speech; SpeechTokenizer and PhoenixCodec use LibriSpeech test-clean. Absolute scores are not used to rank codecs across these different materials.

The database snapshot contains 35 completed non-QA sessions. Each reported codec has 20 panels across $\bar{d} \in \{2, 4\}$, with seven ratings per panel, yielding 140 ratings per method per codec. Sessions contain practice trials, and no panel is repeated within a session. Session identifiers are not a verification of unique participant identities. Repeated ratings within a session must not be treated as independent listeners.

F.2 Listening Scores and Interpretation

All completed non-QA sessions are retained without post-hoc listener exclusions. Table 22 gives the reported codec subset; differences are calculated before rounding. These are descriptive means, not tests of statistically significant perceptual improvement.

Table 22: Mean listening scores across $\bar{d} \in \{2, 4\}$; 140 ratings per method per codec.

Codec	Ref	Proposal	Δ
DAC	44.57	47.04	+2.47
SpeechTokenizer	48.79	50.97	+2.18
PhoenixCodec	51.63	54.14	+2.51

F.3 Response Quality

The study-wide database audit found no missing completed submissions, out-of-range ratings, or insufficient recorded playback coverage under its technical checks. However, hidden-reference scores were frequently below 90. Playback coverage does not establish attentive listening, and no post-hoc quality-based session exclusion is applied to the primary means. The archived study-wide diagnostic is not presented as a newly computed diagnostic restricted to the three reported codecs. These response-quality concerns limit the strength of the listening evidence. Pooled auxiliary preference counts are not used to support claims about the three-codec selector.